CLAIMS

1. A computer system for executing instructions having assigned guard indicators, the system comprising:

instruction supply circuitry;

a plurality of parallel execution units for receiving respective instructions from the instruction supply circuitry, each of the instructions having a respective guard indicator selected from a set of guard indicators common to the plurality of execution units, at least one of the plurality of execution units including a master guard value store containing a master representation of current values for the guard indicators in the set of guard indicators; and

guard value transfer circuitry operable to transfer a guard value from the master store to another of the execution units in response to a sendguard instruction being executed in said one execution unit;

wherein the instruction supply circuitry comprises a main instruction queue for holding instructions to be supplied to the parallel execution units and a subsidiary instruction queue for holding sendguard instructions, the subsidiary queue having priority access to execution pipelines to avoid unnecessary delays for execution of sendguard instructions.

2. The computer system of claim 1, wherein the execution unit includes circuitry for executing guard value modifying instructions which are held in the main instruction queue and for updating said master guard value store with any modified guard value.

3. The computer system of claim 2, wherein the instruction supply circuitry comprises a control unit common to each parallel execution unit, said control unit including queue checking circuitry which is operable to check, before each sendguard instruction is supplied to the execution pipelines, that no earlier guard value modifying instructions affecting a guard value requested by the sendguard instruction is still waiting in the main instruction queue.

4. The computer system of claim 2, wherein each execution unit comprises a plurality of pipelined stages with synchronized pipelined cycles for each of the execution

units and the circuitry for executing guard value modifying instructions is located in a pipelined stage downstream of an earlier pipelined stage operable to execute sendguard instructions.

5. The computer system of claim 4, further comprising switching circuitry in an earlier pipelined stage for supplying a guard value which has been modified by a guard value modifying instruction in a subsequent pipelined stage in a machine cycle responsive to a sendguard instruction in the earlier pipelined stage in a same machine cycle.

6. The computer system of claim 1, wherein each execution unit comprises a plurality of pipelined stages with synchronized pipelined cycles for each of the execution units.

7. The computer system of claim 1, wherein each execution unit comprises first, second and third pipelined stages wherein the first and second pipelined stages contain circuitry for executing sendguard instructions, and the third pipelined stage includes circuitry for executing guard value modifying instructions.

8. The computer system of claim 7, further comprising switching circuitry for selecting, responsive to a sendguard instruction, whether a guard value held in the master guard value store or a guard value just modified by a guard value modifying instruction is dispatched via the guard value transfer circuitry.

9. The computer system of claim 1, wherein said execution unit includes execution pipelines providing access to a data memory, said pipelines including a first set of pipelines for use in executing instructions needed for memory access operations and a second set of pipelines arranged to carry out arithmetic operations, thereby providing decoupling of memory access operations from arithmetic operations.

10. The computer system of claim 9, wherein two parallel pipelines are provided for arithmetic operations and access a common set of data registers including said master guard value store.

11. The computer system of claim 1, wherein said master guard value store comprises a register file.

12. A method of executing instructions in a computer system, said instructions having assigned guard indicators, the method comprising:

supplying a plurality of instructions to a plurality of parallel execution units, each instruction having a respective guard indicator selected from a set of guard indicators common to the plurality of parallel execution units;

holding a master set of current values for the guard indicators in one execution unit; and

effecting a transfer of the master guard value from said one execution unit in response to a sendguard instruction being executed in said one execution unit;

wherein instructions to be supplied to the parallel execution units, except for said sendguard instructions, are held in a main instruction queue; and

wherein said sendguard instructions are held in a subsidiary queue having priority access to execution pipelines to avoid unnecessary delays for execution of sendguard instructions.

13. The method of claim 12, including executing instructions to modify said guard values wherein execution of an instruction to modify a guard value updates said master value, said guard value modifying instructions being held in the main instruction queue.

14. The method of claim 13, further comprising a step of checking, before each sendguard instruction is supplied to the execution pipelines, that no earlier guard value modifying instruction affecting the guard value requested by the sendguard instruction is still waiting in the main instruction queue.

15. A pipelined execution unit for a computer system which comprises at least two pipelined stages, with an earlier one of the pipelined stages including sendguard circuitry responsive to a sendguard instruction to dispatch a guard value for a guard indicator defined in the sendguard instruction, and a later one of the pipelined

stages including guard value modifying circuitry for executing guard value modifying instructions which cause the value of a guard indicator to be modified, the pipelined execution unit further comprising:

a master guard value store for holding a master representation of current values for guard indicators;

means for determining any dependencies between a sendguard instruction in the earlier pipelined stage and a guard modifying instruction in the later pipelined stage; and

switching circuitry for selecting when a send guard instruction is executed and, responsive to any such dependencies, whether a guard value held in the master guard value store or a guard value just modified by the guard value modifying circuitry is to be dispatched.

16. The pipelined execution unit of to claim 15, which comprises first, second and third pipelined stages wherein the first and second pipelined stages each contain such circuitry for executing sendguard instructions, and the third pipelined stage includes such circuitry for executing guard value modifying instructions.

17. The pipelined execution unit of claim 15, further including a data write-back stage for writing back the results of execution of an instruction into a data store.

18. A method of executing instructions in a pipelined execution unit, each instruction having a respective guard indicator selected from a set of guard indicators and

execution of the instructions being predicated on values of the guard indicators, wherein the instructions include a sendguard instruction which, when executed, causes transfer of a guard value from the pipelined execution unit and a guard value modifying instruction which modifies the value of a guard indicator, the method comprising:

supplying instructions to the pipelined execution unit including said sendguard instructions and said guard value modifying instructions;

checking dependencies between guard value modifying instructions supplied to the pipelined execution unit earlier than a sendguard instruction relating to a same guard indicator; and

supplying the guard value of the guard indicator requested in the sendguard instruction selectively from a master guard value store or guard value modifying circuitry in dependence on results of said checking step, so as to ensure that the guard value of a guard indicator which is dispatched responsive to a sendguard instruction is correct in relation to any earlier guard value modifying instructions in the pipelined execution unit.

19. A pipelined execution unit for a computer system which comprises at least two pipelined stages, with an earlier one of the pipelined stages including sendguard circuitry responsive to a sendguard instruction to dispatch a guard value for a guard indicator defined in the sendguard instruction, and a later one of the pipelined stages including guard value modifying circuitry for executing guard value modifying instructions which cause the guard value of the guard indicator to be modified, the pipelined execution unit further comprising:

a master guard value store for holding a master representation of current values for guard indicators;

a dependency determiner for determining any dependencies between a sendguard instruction in the earlier pipelined stage and a guard modifying instruction in the later pipelined stage; and

switching circuitry for selecting when a send guard instruction is executed and responsive to any such dependencies whether a guard value held in the master guard value store or a guard value just modified by the guard value modifying circuitry is to be dispatched.

20. A method of handling guarded instructions within a processor, the processor including several processing units associated respectively with FIFO-type memory means for sequentially storing the respective guarded instructions which are intended for corresponding processing units, a first unit containing a guard-indications register, in which method issuing of a guarded instruction to the memory means of a second processing unit causes issuing to the memory means of the first processing unit of a transmission instruction intended to have a value of a guard indication associated with said guarded instruction transmitted to the second unit, wherein:

the memory means of the first unit includes a first FIFO-type memory, and a

second FIFO-type memory separate from the first one;

each transmission instruction is stored in the first memory and all other instructions intended for the first unit are stored in the second memory;

a transmission instruction having reached a head of the first memory is extracted from the first memory, if no modifying instruction which is earlier in time and intended to modify the value of the guard indication associated with the transmission instruction is present in the second memory; and

in the presence of such an earlier, modifying instruction, the transmission instruction is extracted from the first memory only after the modifying instruction has been extracted from the second memory.

21. The method of claim 20, wherein:

every time an instruction is extracted from the second memory a read counter is incremented;

every time an instruction is stored in the second memory, a write counter is incremented;

every time an instruction modifying the value of a guard indication is stored in the second memory, a current value of the write counter is stored in memory; and

in that determining of a still-present character of the modifying instruction in the second memory takes account of a result of a comparison of the current value of the write counter with the current value of the read counter.

22. The method of claim 21, wherein:

the write counter and the read counter have an identical size which is equal to a depth of the second memory;

an overflow bit changing value every time a corresponding counter comes back to its initial value is associated with each counter;

every time an instruction modifying the value of a guard indication is stored in the second memory, the current value of the overflow bit of the write counter is likewise stored in memory; and

the determining of the still-present character of the modifying instruction in the second memory also takes into account the result of the comparison of the current value of the overflow bit of the read counter with a memory-stored value of the overflow bit of

the write counter.

23. The method of claim 21, wherein:

every time a transmission instruction is stored in the first memory, without another instruction simultaneously being stored in the second memory, a non-operative instruction is simultaneously stored in the second memory;

every time a transmission instruction is stored a label is associated with said transmission instruction, containing the current value of the write counter which was stored in memory when a last instruction modifying the guard indication associated with said transmission instruction was stored in the second memory; and

a criterion for extraction of the transmission instruction having reached the head of the first memory takes into account the result of the comparison between said memory-stored current value associated with this transmission instruction and the current value of the read counter.

24. The method of claim 22, wherein:

every time a transmission instruction is stored in the first memory, the current value of the overflow bit of the write counter which was stored in memory when the last instruction modifying the guard indication associated with said transmission instruction was stored in the second memory is associated with it, also in its label; and

a criterion for extraction of the transmission instruction having reached the head of the first memory takes into account the result of the comparison between this memory-stored current value associated with this transmission instruction and the current value of the overflow bit of the read counter.

25. A processor, comprising several processing units, associated respectively with FIFO-type memory means for sequentially storing respective instructions which are intended for corresponding units, a first unit containing a guard-indications register, and a central unit able to issue a guarded instruction to the memory means of a second processing unit, and to issue to the memory means of the first processing unit a transmission instruction intended to have a value of a guard indication associated with said guarded instruction transmitted to the second unit, wherein:

the memory means of the first unit includes a first FIFO-type memory and a

second FIFO-type memory separate from the first one; and

the processor includes:

routing means, able to store each transmission instruction in the first memory and all other instructions intended for the first unit in the second memory; and

control means able to extract from the first memory a transmission instruction having reached a head of the first memory if no modifying instruction which is earlier in time and intended to modify the value of the guard indication associated with the transmission instruction is present in the second memory, and

in the presence of such an earlier modifying instruction, to extract the transmission instruction from the first memory only after the modifying instruction has been extracted from the second memory.

26. The processor of claim 25, wherein the control means includes:

a read counter incremented every time an instruction is extracted from the second memory;

a write counter incremented every time an instruction is stored in the second memory;

a set of individual registers associated respectively with the set of guard indications; and

a first control unit able, every time an instruction modifying the value of a guard indication is stored in the second memory, to store the current value of the write counter in a main field of an individual register associated with the guard indication,

a second control unit able to determine a still-present character of the modifying instruction in the second memory, and including comparison means able to compare a content of the main field of the individual register with the current value of the read counter.

27. The processor of claim 26, wherein:

the write counter and the read counter have an identical binary size equal to a depth of the second memory;

an overflow bit changing value every time a corresponding counter comes back to its initial value is associated with each counter;

each individual register further includes an auxiliary, one-bit field;

the first control unit is able, every time an instruction modifying the value of a guard indication is stored in the second memory, also to store a current value of the overflow bit of the write counter in the auxiliary field of a corresponding individual register; and

the second control unit includes auxiliary comparison means able to compare the current value of the overflow bit of the read counter with a content of the auxiliary field.

28. The processor of claim 27, wherein the auxiliary comparison means includes an EXCLUSIVE NOR logic gate.

29. The processor of claim 26, wherein:

each stage of the first memory includes a field which is usable for storage of a transmission instruction, and a first supplementary field;

the control means includes a third control unit able, every time a transmission instruction is stored in a usable field of an input stage of the first memory, to transfer the content of the main field of an individual register associated with the corresponding guard indication into the first supplementary field of the input stage of the first memory, and a fourth control unit able to derive a criterion for extraction of the transmission instruction having reached a header stage of the third memory, and including comparison means able to compare a content of said first supplementary field of the header stage with a current value of the read counter.

30. The processor of claim 29, wherein:

each stage of the first memory further includes a second supplementary field;

the third control unit is able, every time a transmission instruction is stored in the usable field of the input stage of the first memory, to transfer the content of the auxiliary field of the individual register associated with the corresponding guard indication into the second supplementary field of the input stage of the first memory; and

the fourth control unit includes supplementary comparison means able to compare the content of the second supplementary field with the current value of the overflow bit of the read counter.

31. The processor of claim 30, wherein the supplementary comparison means includes an EXCLUSIVE NOR logic gate.